

The Ehrlich-Aberth method for palindromic matrix polynomials represented in the Dickson basis

Luca Gemignani^a, Vanni Noferini^{a,*}

^a*Department of Mathematics, University of Pisa
Largo Bruno Pontecorvo 5, 56127 Pisa, Italy*

Abstract

An algorithm based on the Ehrlich-Aberth root-finding method is presented for the computation of the eigenvalues of a T-palindromic matrix polynomial. A structured linearization of the polynomial represented in the Dickson basis is introduced in order to exploit the symmetry of the roots by halving the total number of the required approximations. The rank structure properties of the linearization allow the design of a fast and numerically robust implementation of the root-finding iteration. Numerical experiments that confirm the effectiveness and the robustness of the approach are provided.

AMS classification: 65F15

Keywords: Generalized eigenvalue problem, root-finding algorithm, rank-structured matrix

1. Introduction

The design of effective numerical methods for solving structured generalized eigenvalue problems has recently attracted a great deal of attention. Palindromic matrix polynomials arise in many applications [20]. An $n \times n$ matrix polynomial of degree k $P(z) = \sum_{i=0}^k A_i z^i$, $A_k \neq 0$, $A_i \in \mathbb{C}^{n \times n}$, $0 \leq i \leq k$, is said to be T-palindromic if $A_i^T = A_{k-i}$ for $i = 0, \dots, k$. It is well-known [20], [21] that the palindromic structure induces certain spectral symmetries: in particular if $\lambda \neq 0$ is an eigenvalue of $P(z)$ then $1/\lambda$ is also an eigenvalue of $P(z)$. Numerical solution methods are generally asked to preserve these symmetries.

The customary approach for polynomial eigenproblems consists in two steps: First $P(z)$ is linearized into a matrix pencil $L(z) = zX + Y$, $X, Y \in \mathbb{C}^{nk \times nk}$, and then the eigenvalues of $L(z)$ are computed by some iterative solver. The usual choice of the matrix QZ algorithm applied to a companion linearization [12] of $P(z)$ is implemented in the Matlab function *polyeig*. An alternative solver

*Corresponding author.

Email addresses: gemignan@dm.unipi.it (Luca Gemignani),
noferini@mail.dm.unipi.it (Vanni Noferini)

based on the Ehrlich-Aberth root finding algorithm is proposed in [5] for dealing with certain structured linearizations. Specifically the method of [5] is designed to solve generalized tridiagonal eigenvalue problems but virtually, as shown below, it can be extended to several other rank structures. A generalization for tridiagonal quadratic eigenvalue problems is presented in [24]. A similar strategy using Newton's iteration directly applied to compute the zeros of $P(z)$ is pursued in [11].

Modified methods for palindromic eigenproblems which are able to preserve their spectral symmetries have been proposed in several papers. The construction of T-palindromic linearizations of palindromic eigenproblems is the subject of [19] and [7], whereas numerical methods based on matrix iterations have been devised in [23], [18], [25] and [22] for computing the eigenvalues of these linearizations by maintaining the palindromic structure throughout the computation. To date, however, the authors are not aware of any specific adaptation of the root-finding based methods to palindromic structures.

The contribution of this paper is to fill the gap by developing a root finder specifically suited for T-palindromic matrix polynomials, with particular emphasis on the case of large degree. T-palindromic polynomials of large even degree arise as truncation of Fourier series in several applications such as spectral theory, filtering problems, optimal control and multivariate discrete time series prediction [27].

The polynomial root-finding paradigm is a flexible, powerful and quite general tool for solving both structured and unstructured polynomial eigenproblems. In its basic form it proceeds in four steps:

1. The matrix polynomial is represented in some convenient polynomial basis.
2. The transformed polynomial is linearized.
3. The linearization is reduced in the customary Hessenberg-triangular form.
4. A root-finding method is applied for approximating the eigenvalues of the (reduced) pencil.

This scheme has some degrees of freedom concerning the choice of the polynomial basis at step 1 and the choice of the linearization at step 2 which can be used to exploit both structural and root properties of the matrix polynomial. The complexity heavily depends on the efficiency of the polynomial zero-finding method applied to the determinant of the pencil. Steps 2 and 3 are optional but can substantially improve the numerical and computational properties of the method. Some caution should be used at step 1 since the change of the basis could modify the spectral structure of the matrix polynomial. The key idea we propose for the implementation of step 4 is the use of the Jacobi formula [13]. We emphasize that, although in this paper we focus on palindromics and on a version of the method that is able to extract the palindromic spectral structure, this strategy may be used to address the most general case of an unstructured matrix polynomial eigenproblem, for instance by applying it to the companion linearization. An analysis of the application of the method to a generic matrix polynomial will appear elsewhere.

In this paper we consider the polynomial root-finding paradigm for solving T-palindromic eigenproblems. In particular, we address the main theoretical and computational issues arising at steps 1, 2 and 4 of the previous scheme applied to T-palindromic matrix polynomials, and also we indicate briefly how to carry out the reduction at step 3. The proposed approach relies upon the representation and manipulation of T-palindromic matrix polynomials in a different degree-graded polynomial basis $\{\phi_j(y)\}$, namely the Dickson basis, satisfying a three-term recurrence relation and defined by $\phi_0(y) = 2$, $\phi_1(y) = y$ and $y\phi_j(y) = \phi_{j+1}(y) + \phi_{j-1}(y)$ for $j = 1, 2, \dots$. For the given T-palindromic polynomial $P(z)$ of degree $k = 2h$ we determine a novel polynomial $M(y) = \sum_{j=0}^{h+1} M_j \phi_j(y)$, $M_j \in \mathbb{C}^{2n \times 2n}$, $0 \leq j \leq h+1$, $y = z + z^{-1}$, with the property that if λ and λ^{-1} are two distinct (i.e. $\lambda \neq \pm 1$) finite semi-simple eigenvalues of $P(z)$ with multiplicity ℓ , then $\mu = \lambda + \lambda^{-1}$ is a semi-simple eigenvalue for $M(y)$ with multiplicity 2ℓ . Moreover, we find that

$$g(y) = \det(M(y)) = [\det(z^{-h}P(z))]^2 = p(y) \cdot p(y),$$

where $p(y)$ is a polynomial of degree nh at most.

Solving the algebraic equation $p(y) = 0$ is at the core of our method for T-palindromic eigenproblems. Our computational experience in polynomial root-finding indicates that the Ehrlich-Aberth method [1], [9] for the simultaneous approximation of polynomial zeros realizes a quite good balancing between the quality of convergence and the cost per iteration. The main requirements for the effective implementation of the Ehrlich-Aberth method are both a fast, robust and stable procedure to evaluate the Newton correction $p(y)/p'(y)$ and a reliable criterion to stop the iteration. Concerning the first issue it is worth noting that $p(y)/p'(y) = 2g(y)/g'(y)$ and, therefore, the computation immediately reduces to evaluating the Newton correction of $g(y)$. A suitable structured linearization $L(y)$ of $M(y)$ can be obtained following [2] which displays a semiseparable structure. In this way, in view of the celebrated Jacobi Formula[13] $g'(y)/g(y) = \text{trace}((L(y))^{-1}L'(y))$, the Newton correction can be evaluated by performing a QR factorization of $L(y)$, say $L(y) = Q(y) \cdot R(y)$, at low computational cost and fulfilling the desired requirements of robustness and stability. Also, since $\|(L(y))^{-1}\|_2 = \|(R(y))^{-1}\|_2$ for $y \notin \text{spec}(L(y))$ we obtain at no additional cost a reliable stop condition based on an estimate of the backward error given by Higham and Higham [14].

If k , the degree of the matrix polynomial, is large with respect to n , the size of its matrix coefficients, our approach looks appealing since with a smart choice of the starting points it needs $\mathcal{O}(n^4k + n^3k^2)$ operations, whereas the QZ method makes use of $\mathcal{O}(n^3k^3)$ operations. The unpleasant factor n^4 in our cost estimate depends on the block structure of the linearization used in our current implementation and can in principle be decreased by performing the preliminary reduction of the linearization in Hessenberg-triangular form as stated at step 3 of the basic scheme. The reduction can be carried out by a structured method exploiting the semiseparable structure of the block linearization to compute a rank-structured Hessenberg-triangular linearization.

Incorporating the structured method in our implementation would finally lead to a fast method that outperforms the QZ algorithm for large degrees and is comparable in cost for small degrees.

The paper is organized as follows. The theoretical properties of the considered linearizations of T-palindromic matrix polynomials expressed in the Dickson basis are investigated in Section 2 and 3. The derivation of the proposed eigenvalue method for T-palindromic eigenproblems is established in Section 4 and 5. The complete algorithm is described in Section 6. Numerical experiments are presented in Section 7 to illustrate the robustness of our implementation and to indicate computational issues and possible improvements of our algorithm compared with other existing methods. Finally, conclusion and future work are discussed in Section 8.

2. Theoretical preliminaries on polynomial bases linearizations

This preparatory section recalls some basic definitions, background facts and notations used throughout the paper.

For $j = 0, \dots, k$ let $P_j \in \mathbb{C}^{n \times n}$, $P_k \neq 0$, be constant matrices and consider the matrix polynomial $P = P(\lambda) = \sum_{j=0}^k P_j \lambda^j$. The generalized polynomial eigenproblem (PEP) associated to $P(\lambda)$ is to find an eigenvalue λ_0 and a corresponding nonzero eigenvector \mathbf{x}_0 satisfying

$$P(\lambda_0)\mathbf{x}_0 = \mathbf{0}. \quad (1)$$

In this paper, we will always suppose that $P(\lambda)$ is *regular*, i.e. its determinant does not identically vanish.

A *linearization* of $P(\lambda)$ is defined as a pencil $L(\lambda) = \lambda X + Y$, with $X, Y \in \mathbb{C}^{kn \times kn}$, such that there exist unimodular polynomial matrices $E(\lambda)$ and $F(\lambda)$ for which

$$E(\lambda)L(\lambda)F(\lambda) = \begin{pmatrix} P(\lambda) & 0 \\ 0 & I_{(k-1)n} \end{pmatrix}.$$

Moreover, if one defines the reversal of a matrix polynomial as $\text{rev}(P) := \lambda^k \sum_j P_j \lambda^{-j}$, the linearization is said to be *strong* whenever $\text{rev}(L) = \lambda Y + X$ is a linearization of $\text{rev}(P)$.

Following the work of Mackey, Mackey, Mehl and Mehrmann [21], in the paper [15] Higham, Mackey, Mackey and Tisseur study the two (right and left) *ansatz vector linearization spaces*: having introduced the vector $\mathbf{\Lambda} := (1, \lambda, \dots, \lambda^{k-1})^T$, these spaces are defined as follows:

$$\hat{\mathcal{L}}_1 := \{L = \lambda X + Y : \exists \mathbf{v} \in \mathbb{C}^k \text{ s.t. } L \cdot (\mathbf{\Lambda} \otimes I_n) = \mathbf{v} \otimes P\} \quad (2)$$

$$\hat{\mathcal{L}}_2 := \{L = \lambda X + Y : \exists \mathbf{w} \in \mathbb{C}^k \text{ s.t. } (\mathbf{\Lambda}^T \otimes I_n) \cdot L = \mathbf{w}^T \otimes P\}. \quad (3)$$

It is shown in [21] that almost every pencil in these spaces is a linearization, while in [15] two binary operations on block matrices, called column shifted sum and row shifted sum, are first introduced and then used to characterize the above defined spaces.

On the other hand, in [2] Amiraslani, Corless and Lancaster consider linearizations of a matrix polynomial expressed in some polynomial bases different than the usual monomial one. Equation (7) in [2] resembles closely the defining equation of $\hat{\mathcal{L}}_2$. The authors themselves stress this analogy, that suggests an extension of the results of [15] to the case of different polynomial bases. Let $\{\phi_i\}_{i=0,\dots,k}$ be a basis for the polynomials of degree less than or equal to k . In [2] degree-graded bases that satisfy a three-terms recurrence relation (for instance, orthogonal polynomials always do so) are considered:

$$\lambda\phi_j(\lambda) = \alpha_j\phi_{j+1}(\lambda) + \beta_j\phi_j(\lambda) + \gamma_j\phi_{j-1}(\lambda). \quad (4)$$

The α_j are obviously linked to the leading-term coefficients of the ϕ_j . Specifically, calling c_j such coefficients, one has that $c_j = \alpha_j c_{j+1}$.

We wish to consider the expansion of the polynomial $P(\lambda)$ in this basis:

$$P(\lambda) = \sum_{j=0}^k A_j \phi_j(\lambda). \quad (5)$$

We introduce the vector

$$\Phi := (\phi_0(\lambda), \phi_1(\lambda), \dots, \phi_{k-1}(\lambda))^T.$$

By generalizing the linearizations studied in [2], for each choice of Φ two new ansatz vector linearization spaces can be defined:

$$\mathcal{L}_1 := \{L = \lambda X + Y : \exists \mathbf{v} \in \mathbb{C}^k \text{ s.t. } L \cdot (\Phi \otimes I_n) = c_{k-1} \mathbf{v} \otimes P\}; \quad (6)$$

$$\mathcal{L}_2 := \{L = \lambda X + Y : \exists \mathbf{w} \in \mathbb{C}^k \text{ s.t. } (\Phi^T \otimes I_n) \cdot L = c_{k-1} \mathbf{w}^T \otimes P\}. \quad (7)$$

It is worth noticing that it is not strictly necessary for the new basis to be degree-graded, nor it is to satisfy a three-term recurrence relation. In fact, it is sufficient that $\{\phi_i\}_{i=0,\dots,k-1}$ are linearly independent and have degree less than or equal to $k-1$, so that there exists an invertible basis change matrix B such that $\Phi = B\Lambda$. The basis is degree-graded if and only if B is lower triangular.

In the light of the above definitions it is immediately seen that the main results of [21], [15] remain valid in the case of a more general polynomial basis. In particular the following result holds.

Proposition 1. *Let $L \in \mathcal{L}_1$ (\mathcal{L}_2). The following properties are equivalent:*

- L is a linearization of P
- L is a strong linearization of P
- L is regular

Proof. It is a corollary of Theorem 4.3 of [21]. In fact, any $L \in \mathcal{L}_1$ (resp., \mathcal{L}_2) can be written as $L = c_{k-1} \hat{L} \cdot (B^{-1} \otimes I_n)$ (resp., $L = c_{k-1} (B^{-T} \otimes I_n) \hat{L}$) for some $\hat{L} \in \hat{\mathcal{L}}_1$ (resp., $\hat{\mathcal{L}}_2$). Therefore, L has each of the three properties above if and only if \hat{L} has the corresponding property. \square

This proposition guarantees that almost every (more precisely, all but a closed nowhere dense set of measure zero) pencil in \mathcal{L}_1 (\mathcal{L}_2) is a strong linearization for P . For a proof, see Theorem 4.7 of [21]. The eigenvectors of L are related to those of P . More precisely, $(\lambda, \Phi \otimes \mathbf{x})$ is an eigenpair for L if and only if (λ, \mathbf{x}) is an eigenpair for P . Moreover, if L is a linearization then every eigenvector of L is of the form $\Phi \otimes \mathbf{x}$ for some eigenvector \mathbf{x} of P . A similar recovery property holds for the left ansatz vector linearizations. These properties can be simply proved as in Theorems 3.8 and 3.14 of [21], that demonstrate them for the special case $\Phi = \mathbf{A}$.

For the numerical treatment of palindromic generalized eigenproblems a crucial role is played by the so-called Dickson basis [8] $\{\phi_i\}_{i \geq 0}$ defined by

$$\begin{cases} \phi_0(y) = 2 \\ \phi_1(y) = y \\ \forall j \geq 1, \quad y\phi_j(y) = \phi_{j+1}(y) + \phi_{j-1}(y). \end{cases} \quad (8)$$

If we consider the mapping $y := \lambda + \lambda^{-1}$ (which we will refer to as the Dickson transformation or the Dickson change of variable) then $\lambda^j + \lambda^{-j} = \phi_j(y)$ for $j = 0, 1, \dots$. For $\lambda = e^{i\alpha}$, we obtain that $\phi_j(y) = 2 \cos(j\alpha)$. From [2] by choosing \mathbf{e}_k as the ansatz vector we find a suitable strong linearization of $P(\lambda)$ represented as in (5):

$$\begin{pmatrix} I_n & & & & & \\ & I_n & & & & \\ & & \ddots & & & \\ & & & I_n & & \\ & & & & A_k & \end{pmatrix} \lambda + \begin{pmatrix} 0 & -2I_n & & & & \\ -I_n & 0 & -I_n & & & \\ & -I_n & 0 & -I_n & & \\ & & \ddots & \ddots & \ddots & \\ & & & -I_n & 0 & -I_n \\ A_0 & A_1 & \dots & A_{k-3} & A_{k-2} - A_k & A_{k-1} \end{pmatrix}. \quad (9)$$

In the next section we study the spectral modifications induced by the Dickson change of variable that provide the basic link between palindromic matrix polynomials and matrix polynomials expressed in the Dickson basis.

3. Preservation of Jordan structure in the Dickson transformation

Let us recall that if λ_0 is an eigenvalue of $P(\lambda)$ then the set $\{\mathbf{x}_j\}$, $j = 0, \dots, \ell$ is a Jordan chain of length $\ell + 1$ if $\mathbf{x}_0 \neq \mathbf{0}$ and the following relations hold [12]:

$$\sum_{i=0}^m \frac{P^{(m-i)}(\lambda_0)}{(m-i)!} \mathbf{x}_i = \mathbf{0}, \quad m = 0, \dots, \ell,$$

where $P^{(k)}(\lambda_0)$ denotes the k -th derivative of $P(\lambda)$ evaluated at $\lambda = \lambda_0$. The case $m = 0$ corresponds to the definition of an eigenvector. The notion of a Jordan chain can be extended to any matrix function $F : \mathbb{C} \rightarrow \mathbb{C}^{n \times n}$ whose determinant vanishes at λ_0 , as long as $F(\lambda)$ is analytic in a neighborhood of λ_0 . In particular, the case of Laurent polynomials is important for our investigations. If the principal part of a Laurent polynomial $L(\lambda)$ is a polynomial of degree k

in $1/\lambda$, then $P(\lambda) = \lambda^k L(\lambda)$ is a polynomial. The following lemma relates the Jordan chains of the two. The proof is a straightforward application of the product differentiation rule.

Lemma 1. *Let $L(\lambda)$ be a (Laurent) polynomial and $P(\lambda) = \lambda^k L(\lambda)$ for some natural number k . Then the set $\{\mathbf{x}_j\}$ is a Jordan chain of length $\ell + 1$ for $P(\lambda)$ associated to the eigenvalue $\lambda_0 \neq 0$ if and only if $\{\mathbf{x}_j\}$ is a Jordan chain of length $\ell + 1$ for $L(\lambda)$ associated to the same eigenvalue.*

Roughly speaking, Lemma 1 makes us able to switch between regular and Laurent polynomials without worrying about changes in eigenvalues and generalized eigenvectors. Actually, this result can be slightly generalized with the next lemma, which is just an adaptation of a well-known result in [12] for the case where the four matrix functions that we are going to consider are polynomials. In order to prove the lemma, we recall [12] that a vector polynomial $\phi(\lambda)$ is called a root polynomial of order $\ell + 1$ corresponding to λ_0 for the matrix polynomial $P(\lambda)$ if the following conditions are satisfied:

$$\begin{cases} \phi(\lambda_0) \neq \mathbf{0}; \\ \lambda_0 \text{ is a zero of order } \ell + 1 \text{ for } P(\lambda)\phi(\lambda). \end{cases} \quad (10)$$

Obviously a root polynomial of order $\ell + 1$ is defined up to an additive term of the form $(\lambda - \lambda_0)^{\ell+1} \mathbf{v}(\lambda)$ for any suitable vector polynomial $\mathbf{v}(\lambda)$. It is possible to prove that $\phi(\lambda) = \sum_{j=0}^{\ell} (\lambda - \lambda_0)^j \phi_j + (\lambda - \lambda_0)^{\ell+1} \mathbf{v}(\lambda)$ if and only if $\{\phi_j\}$ is a Jordan chain of length $\ell + 1$ for $P(\lambda)$ at $\lambda = \lambda_0$. When $\lambda_0 \neq 0$, thanks to Lemma 1 it is possible to extend the concept to Laurent polynomials: if $L(\lambda)$ is a Laurent polynomial whose singular part has degree k as a polynomial in λ^{-1} , then we say that $\phi(\lambda)$ is a root polynomial for $L(\lambda)$ if it is a root polynomial for $\lambda^k L(\lambda)$.

Lemma 2. *Let $P_1(\lambda)$, $P_2(\lambda)$ be (Laurent) polynomials and $A(\lambda)$, $B(\lambda)$ be two matrix functions with $P_2(\lambda) = A(\lambda)P_1(\lambda)B(\lambda)$. Suppose that an open neighborhood Ω of $\lambda_0 \neq 0$ exists such that all the considered functions are analytic in Ω , and also suppose that both $A(\lambda_0)$ and $B(\lambda_0)$ are invertible. Then λ_0 is an eigenvalue for P_1 if and only if it is an eigenvalue for P_2 , and $\{\mathbf{y}_i\}$ is a Jordan chain of length $\ell + 1$ for P_2 at λ_0 if and only if $\{\mathbf{z}_i\}$ is a Jordan chain of length $\ell + 1$ for P_1 at λ_0 , where $\mathbf{z}_i = \sum_{j=0}^i \frac{B^{(j)}(\lambda_0)}{j!} \mathbf{y}_{i-j}$.*

Proof. If $P_1(\lambda)$ and $P_2(\lambda)$ are classical polynomials then the thesis follows as in the proof of Proposition 1.11 in [12] after having represented $A(\lambda)$ and $B(\lambda)$ by their Taylor series expansions. To deal with the Laurent case, let α and β be the minimal integers such that $Q_1(\lambda) := \lambda^\alpha P_1(\lambda)$ and $Q_2(\lambda) := \lambda^\beta P_2(\lambda)$ are classical polynomials. Just follow the previous proof for $Q_2(\lambda) = \lambda^{\beta-\alpha} A(\lambda) Q_1(\lambda) B(\lambda)$ and apply Lemma 1. \square

We are now in the position to prove a result for the Dickson change of variable $y = \lambda + 1/\lambda$. The following proposition shows that the number of Jordan chains and their length at some eigenvalue y_0 (for the sake of brevity, we shall use the

expression *Jordan structure at y_0*) is related to the Jordan structures at λ_0 and λ_0^{-1} .

Lemma 3. *Let $y(\lambda) = \lambda + \lambda^{-1}$ and let $M(y)$ be a polynomial in y , so that $N(\lambda) := M(y(\lambda))$ is a Laurent polynomial in λ . Let first $y_0 = \lambda_0 + \lambda_0^{-1}$, $\lambda_0 \neq \pm 1$, be a finite eigenvalue of $M(y)$. Then the Jordan structure of $M(y)$ at y_0 is equal to the Jordan structure of $N(\lambda)$ at either λ_0 or $1/\lambda_0$. If on the contrary $\lambda_0 = \pm 1$, then there is a Jordan chain of length ℓ at $M(\pm 2)$ if and only if there is a Jordan chain of length 2ℓ at $N(\pm 1)$.*

Proof. It is obvious that $y_0 \in \mathbb{C}$ is an eigenvalue for $M(y)$ if and only if both λ_0 and λ_0^{-1} are eigenvalues of $N(\lambda)$. Let $M(y) = E(y)D(y)F(y)$, where $D(y) = \text{diag}(d_1(y), \dots, d_n(y))$ is the Smith form ([12],[26]) of $M(y)$. Define $\hat{E}(\lambda) := E(y(\lambda))$, $\hat{D}(\lambda) := D(y(\lambda))$, $\hat{F}(\lambda) := F(y(\lambda))$. If α, β, γ are such that $\tilde{N}(\lambda) = \lambda^{\alpha+\beta+\gamma}N(\lambda)$, $\tilde{E}(\lambda) = \lambda^\alpha \hat{E}(\lambda)$, $\tilde{D}(\lambda) = \lambda^\beta \hat{D}(\lambda)$ and $\tilde{F}(\lambda) = \lambda^\gamma \hat{F}(\lambda)$ are polynomials in λ , then we have the relation $\tilde{N}(\lambda) = \tilde{E}(\lambda)\tilde{D}(\lambda)\tilde{F}(\lambda)$; however, in general $\tilde{D}(\lambda)$ is not the Smith form of $\tilde{N}(\lambda)$. Nevertheless, it has the form $\text{diag}(\lambda^{k_1}\tilde{d}_1(\lambda), \dots, \lambda^{k_n}\tilde{d}_n(\lambda))$ where $k_1 \geq k_2 \geq \dots \geq k_n$ and $\tilde{d}_i(\lambda) = \lambda^{\deg(d_i)}d_i(y(\lambda))$. In other words, the $\tilde{d}_i(\lambda)$ -s are palindromic polynomials with no roots at 0 and such that $\tilde{d}_i(\lambda)$ divides $\tilde{d}_{i+1}(\lambda)$ for $i = 1, \dots, n-1$. Moreover, y_0 is a zero of multiplicity n for $d_i(y)$ if and only if both λ_0 and λ_0^{-1} are zeros of multiplicity n for $\tilde{d}_i(\lambda)$. To reduce $\tilde{D}(\lambda)$ into a Smith form, we proceed by steps working on 2×2 principal submatrices.

In each step, we consider the submatrix $\begin{pmatrix} \lambda^\alpha \tilde{d}_i(\lambda) & 0 \\ 0 & \lambda^\beta \tilde{d}_j(\lambda) \end{pmatrix}$, with $i < j$. If $\alpha \leq \beta$, then do nothing; if $\alpha > \beta$, premultiply the submatrix by $\begin{pmatrix} 1 & \\ -b(\lambda) & 1-b(\lambda) \end{pmatrix}$ and postmultiply it by $\begin{pmatrix} a(\lambda) & -q(\lambda) \\ b(\lambda) & \lambda^{\alpha-\beta} \end{pmatrix}$, where $q(\lambda) = \tilde{d}_j(\lambda)/\tilde{d}_i(\lambda)$ while $a(\lambda)$ and $b(\lambda)$ are such that $a(\lambda)\lambda^\alpha \tilde{d}_i(\lambda) + b(\lambda)\lambda^\beta \tilde{d}_j(\lambda) = \lambda^\beta \tilde{d}_i(\lambda)$; the existence of two such polynomials is guaranteed by Bezout's lemma, since $\lambda^\beta \tilde{d}_i(\lambda)$ is the greatest common divisor of $\lambda^\alpha \tilde{d}_i(\lambda)$ and $\lambda^\beta \tilde{d}_j(\lambda)$. It is easy to check that both matrices are unimodular, and that the result of the matrix multiplications is $\begin{pmatrix} \lambda^\beta \tilde{d}_i(\lambda) & 0 \\ 0 & \lambda^\alpha \tilde{d}_j(\lambda) \end{pmatrix}$. By subsequent applications of this algorithm we thus conclude that the Smith form of $\tilde{D}(\lambda)$ is $\hat{D}(\lambda) = \text{diag}(\lambda^{k_n}\tilde{d}_1(\lambda), \dots, \lambda^{k_1}\tilde{d}_n(\lambda))$.

It follows that the i th invariant polynomial of $M(y)$ has a root of multiplicity n_i at y_0 if and only if the i th invariant polynomial of $\tilde{D}(\lambda)$ has a root of multiplicity n_i at $\lambda_0 \neq \pm 1$ and a root of multiplicity n_i at $1/\lambda_0$. From Lemma 2, the Jordan structures of $\tilde{N}(\lambda)$ are equal to those of $\tilde{D}(\lambda)$. The thesis follows from the properties of the Smith form and from Lemma 1.

Mutatis mutandis, a similar argument can be used to analyze the case of $\lambda = \pm 1$: notice in fact that $(y \pm 2)^k$ is a factor of the i th invariant polynomial of $M(y)$ if and only if $(\lambda \pm 1)^{2k}$ is a factor of the i th invariant polynomial of $\tilde{D}(\lambda)$. \square

4. Application to palindromic polynomials

We will now specialize our analysis to the case of a matrix polynomial with palindromic structure.

Remark 1. In this section, we will only treat the case of even degree palindromic matrix polynomials. Notice in fact that an odd degree palindromic may always be transformed to an even degree palindromic, either by squaring the variable ($\lambda = \mu^2$) or by multiplication by $(\lambda + 1)I_n$. Potentially, both actions may introduce problems: squaring the variable adds an additional symmetry $\{\mu, -\mu\}$ to the spectrum while multiplying by $\lambda + 1$ increases by n the multiplicity of -1 as an eigenvalue.

However, the first issue may be solved, after passing to Laurent form, by the use of the change of variable $z = (\mu + \mu^{-1})^2$. See also Remark 3.

Regarding the latter issue, since one knows that he is adding n times -1 there is no need to compute it: n of the $(n+1)k$ starting points of the Ehrlich-Aberth iteration shall be set equal to -2 , and there they remain with no further corrections. The shortcoming is that the Jordan structure at $\lambda = -1$ changes.

Let $\tilde{P}(\lambda) = \sum_{j=0}^{2k} \tilde{A}_j \lambda^j$ be a polynomial of even degree. By Lemma 1, switching to the Laurent form is not harmful for finite nonzero eigenvalues and the corresponding (generalized) eigenvectors; we can therefore consider its Laurent counterpart

$$P(\lambda) := \sum_{j=-k}^k A_j \lambda^j. \quad (11)$$

Three different kinds of palindromic structure can be defined. We say that the Laurent polynomial is purely palindromic (resp., \star -palindromic, $\star \in \{T, H\}$) if the following relations hold between its matrix coefficients:

$$\begin{cases} \text{Purely palindromic: } A_j = A_{-j}; \\ \star\text{-palindromic: } A_j = A_{-j}^{\star}. \end{cases}$$

It is well-known that the palindromic structure induces certain symmetries of eigenvalues and eigenvectors: in particular if λ_0 is an eigenvalue, \mathbf{x} is a right eigenvector and \mathbf{z}^T is a left eigenvector, then, denoting complex conjugation with the operator $(\cdot)^*$

$$\begin{cases} \text{if P is purely palindromic, } P(\frac{1}{\lambda_0})\mathbf{x} = \mathbf{0}, \quad \mathbf{z}^T P(\frac{1}{\lambda_0}) = \mathbf{0}; \\ \text{if P is } T\text{-palindromic, } P(\frac{1}{\lambda_0})\mathbf{z} = \mathbf{0}, \quad \mathbf{x}^T P(\frac{1}{\lambda_0}) = \mathbf{0}; \\ \text{if P is } H\text{-palindromic, } P(\frac{1}{\lambda_0^*})\mathbf{z}^* = \mathbf{0}, \quad \mathbf{x}^H P(\frac{1}{\lambda_0^*}) = \mathbf{0}. \end{cases}$$

In this paper we are primarily interested in the design of an efficient solver for T -palindromic eigenproblems. A numerical method will be presented in Subsection 4.2. The proposed approach can however be described very easily with purely palindromic polynomials. Thus we first consider this case for the sake of clarity.

4.1. Purely palindromic polynomials

The most obvious way to deal with this kind of palindromicity is via introduction of the change of variable $y = \lambda + \lambda^{-1}$, in order to halve the degree of the polynomial. More explicitly, one can define $Q(y) := P(\lambda(y))$; clearly, the purely palindromic structure of $P(\lambda)$ guarantees that $Q(y)$ is itself a polynomial in the new variable y . The next proposition is a simple application of Lemmas 1 and 3, and it relates eigenvectors and Jordan chains of the two polynomials:

Proposition 2. *When $\lambda_0 \neq \pm 1$, the Jordan structure of $Q(y)$ at the eigenvalue $y_0 = \lambda_0 + \lambda_0^{-1}$ is equal to the Jordan structure of $P(\lambda)$ at either λ_0 or λ_0^{-1} . If $\lambda_0 = \pm 1$, $Q(y)$ has a Jordan chain of length ℓ at $y_0 = \pm 2$ if and only if $P(\lambda)$ has a Jordan chain of length 2ℓ at $\lambda_0 = \pm 1$.*

In particular, the eigenvectors of $Q(y)$ at y_0 are exactly the same of the eigenvectors of $P(\lambda)$ at λ_0 (or equivalently at λ_0^{-1} , since they are the same).

Albeit very attractive, from a numerical point of view this trick is not very suitable as soon as one considers a high degree polynomial. In fact, the matrix coefficients of $Q(y)$ need to be computed as linear combinations of the ones of $P(\lambda)$. Since the powers of a binomial are involved, the coefficients of these linear combinations would exponentially grow with the polynomial degree. To circumvent this difficulty, we shall make use of the Dickson polynomials (8). The polynomial $Q(y)$ is readily expressed in terms of the $\phi_j(y)$ s since in the Dickson basis the coefficients are just the old ones and therefore no computation at all is needed, namely,

$$Q(y) = \frac{A_0}{2}\phi_0 + \sum_{j=1}^k A_j \phi_j(y). \quad (12)$$

The associated linearization (9) has several computational advantages with respect to other customary linearizations of $P(\lambda)$. Its size is nk versus $2nk$, the spectral symmetries are preserved and, moreover, the linearization displays a semiseparable structure. More precisely, it is of the form $D_0 + D_1 y$ where D_1 is identity plus low rank while D_0 is Hermitian plus low rank. This kind of structure is preserved under the QZ algorithm and it may be exploited for the design of an efficient and numerically robust root-finder applied to the algebraic equation $\det Q(y) = 0$.

4.2. T-palindromic polynomials

Consider now a T-palindromic polynomial of even degree $2k$. We will suppose once more that neither 0 nor ∞ are eigenvalues, so that we can divide by λ^k and consider the Laurent form $P(\lambda)$, which is a T-palindromic Laurent polynomial of degree k both in λ and in λ^{-1} . Since the symmetry $\lambda \leftrightarrow \lambda^{-1}$ is still present in the spectrum, we expect that the Dickson basis may still play a role. However, unlike the purely palindromic case, it is not possible to directly express a T-palindromic polynomial as a polynomial in the variable y . In fact,

splitting $P(\lambda)$ as the sum of its symmetric part and its skew-symmetric part we obtain that

$$P(\lambda) = A_0 + \sum_{j=1}^k \left[\frac{A_j + A_j^T}{2} (\lambda^j + \lambda^{-j}) + \frac{A_j - A_j^T}{2} (\lambda^j - \lambda^{-j}) \right]. \quad (13)$$

If we introduce the new variables $y := \lambda + \lambda^{-1}$ and $w := \lambda - \lambda^{-1}$, then $P(\lambda)$ can be expressed as a bivariate polynomial in w and y which is always linear in w , that is,

$$Q(y, w) = P(\lambda(y, w)) =: B(y) + wC(y).$$

The property follows from (13) by substituting

$$\lambda^j + \lambda^{-j} = \phi_j(y), \quad \lambda^j - \lambda^{-j} = w \left(\frac{1 + (-1)^{j+1}}{2} + \sum_{\ell=1}^{\lfloor j/2 \rfloor} \phi_{j-2\ell+1}(y) \right), \quad j \geq 1.$$

Notice moreover that $B(y)$ is a symmetric polynomial (that is to say, every matrix coefficient is symmetric), $C(y)$ is skew-symmetric, and the operation of transposition corresponds to changing the sign of w , that is,

$$Q^T(y, w) = P^T(\lambda(y, w)) = B(y) - wC(y).$$

In principle one may think of treating $Q(y, w)$ with available techniques for the bivariate eigenvalue problem (see e.g. [16] and references therein), but actually y and w are not independent. They are related by the trigonometric dispersion relation $w^2 = y^2 - 4$. This suggests that it is possible to obtain a univariate polynomial by doubling the dimensions of the matrix coefficients. Let us define

$$M(y) = \begin{pmatrix} B(y) & w^2 C(y) \\ C(y) & B(y) \end{pmatrix}.$$

Then $M(y)$ is a polynomial in y of degree $k + 1$ at most. Moreover, it has the following property: if λ_0 and λ_0^{-1} are two distinct (i.e. $\lambda_0 \neq \pm 1$) finite semisimple eigenvalues of $P(\lambda)$ with multiplicity m , then $y_0 = \lambda_0 + \lambda_0^{-1}$ is a semisimple eigenvalue for $M(y)$ with multiplicity $2m$. To see this, notice first that

$$M(y) = \text{diag}(\sqrt{w}I_n, 1/\sqrt{w}I_n) \begin{pmatrix} B(y) & wC(y) \\ wC(y) & B(y) \end{pmatrix} \text{diag}(1/\sqrt{w}I_n, \sqrt{w}I_n)$$

and

$$\begin{pmatrix} B(y) & wC(y) \\ wC(y) & B(y) \end{pmatrix} = \frac{1}{2} \begin{pmatrix} I_n & -I_n \\ I_n & I_n \end{pmatrix} \begin{pmatrix} Q(y, w) & 0 \\ 0 & Q^T(y, w) \end{pmatrix} \begin{pmatrix} I_n & I_n \\ -I_n & I_n \end{pmatrix}.$$

Hence, we find that

$$M(y) = E(w) \begin{pmatrix} Q(y, w) & 0 \\ 0 & Q^T(y, w) \end{pmatrix} E^{-1}(w), \quad E(w) := \begin{pmatrix} \sqrt{\frac{w}{2}} & -\sqrt{\frac{w}{2}} \\ \sqrt{\frac{1}{2w}} & \sqrt{\frac{1}{2w}} \end{pmatrix} \otimes I_n.$$

Since, as long as $E(w)$ is defined (that is to say $w \neq 0, \infty$ or $\lambda \neq 0, \pm 1, \infty$), $\det(E(w)) = 1$ then

$$\det(M(y)) = [\det(Q(y, w))]^2, \quad \forall (y, w) \in \mathbb{C} \times \mathbb{C}. \quad (14)$$

Therefore, λ_0 has algebraic multiplicity m for $P(\lambda)$ if and only if y_0 has algebraic multiplicity $2m$ for $M(y)$. This gives the factorization

$$\det(M(y)) = p(y) \cdot p(y), \quad (15)$$

for a suitable polynomial $p(y)$ having the zero y_0 of multiplicity m . Concerning eigenvectors, if λ_0 is semisimple, then let \mathbf{x}_j (resp. \mathbf{z}_j), $j = 1, \dots, m$ be the eigenvectors for $P(\lambda)$ (resp. $P^T(\lambda)$) corresponding to λ_0 : it can be easily checked that $\{(w_0 \mathbf{x}_j^T, \mathbf{x}_j^T)^T, (-w_0 \mathbf{z}_j^T, \mathbf{z}_j^T)^T\}$, where $w_0 = \lambda_0 + \lambda_0^{-1}$, are two linearly independent eigenvectors for $M(y)$ corresponding to y_0 . Thus, geometric multiplicity is also $2m$. Indeed, something more can be said in the more general case of Jordan chains.

Proposition 3. *Let $y_0 = \lambda_0 + \lambda_0^{-1}$ be an eigenvalue of $M(y)$ so that λ_0 and λ_0^{-1} are eigenvalues for $P(\lambda)$. If $\lambda_0 \neq 0, \pm 1, \infty$ then the Jordan structure of $M(y)$ at y_0 is equal to the union of the Jordan structures of $P(\lambda)$ at λ_0 and at λ_0^{-1} .*

Proof. Since $P(\lambda)$ is T-palindromic, it is clear that the Jordan structure of

$$R(\lambda) := \begin{pmatrix} P(\lambda) & 0 \\ 0 & P^T(\lambda) \end{pmatrix}$$

at either λ_0 or λ_0^{-1} is the union of the Jordan structures of $P(\lambda)$ at λ_0 and at λ_0^{-1} . Define

$$N(\lambda) := M(y(\lambda)) = E(w(\lambda))R(\lambda)E^{-1}(w(\lambda)).$$

The matrix function $E(w)$, defined in the previous page, is analytic everywhere in the w complex plane but on a branch semiline passing through the origin. Since by hypothesis $w_0 \neq 0$, the branch cut can be always chosen in such a way that $E(w)$ is analytic in a neighborhood of $w_0 = \lambda_0 - \lambda_0^{-1}$, and thus $E(w(\lambda))$ is analytic in a neighborhood of λ_0 . Then we can apply Lemma 2 to conclude that the Jordan structures of $M(\lambda)$ and $R(\lambda)$ are the same. Application of Lemma 3 completes the proof. \square

Remark 2. Another remarkable property of $M(y)$ is that its coefficients are all skew-Hamiltonian, that is to say they can be written as JK where $J = \begin{pmatrix} 0 & -I \\ I & 0 \end{pmatrix}$ and K is some skew-symmetric matrix. This link between T-palindromic and skew-Hamiltonian polynomials is interesting because it may shed more light on the relation between several polynomial structures. It is known that one can easily transform a palindromic polynomial to an even polynomial by a Cayley transformation, and then to a Hermitian polynomial via a multiplication by i (if one started from a real polynomial) or to a symmetric polynomial by squaring the matrix coefficients. On the other hand, Hamiltonian polynomials can lead to

skew-Hamiltonian polynomials by squaring each coefficients, and multiplication by J sends a skew-Hamiltonian polynomial to a skew-symmetric polynomial. The Dickson change of variable, followed by doubling the dimension, is able to map T-palindromic polynomials of even degree to a subset of skew-Hamiltonian polynomials. Unlike some of the other mentioned maps, this is not a bijection between two classes of structured polynomials, because what is obtained is actually a subset of skew-Hamiltonian polynomials. In fact, since the north-west and south-east coefficients of $M(y)$ are the coefficients of $B(y)$ they must be symmetric and there is a relation between the north-east and south-west coefficients of $M(y)$. However, a deeper investigation on this subject is needed in the future.

Remark 3. Notice that a similar technique can be applied to *even/odd* matrix polynomials, that is polynomials whose coefficients alternate between symmetric and skew-symmetric matrices. In this case, one can apply the transformation $z = \lambda^2$ and use algebraic manipulations, akin to the ones described for the T-palindromic case, in order to build a new polynomial in z with double dimensions.

In the case of an odd-degree T-palindromic polynomial, the substitutions $\lambda = \mu^2$ and $y = \mu + \mu^{-1}$ lead to an $M(y)$ such that $\begin{pmatrix} 0 & I \\ I & 0 \end{pmatrix} \cdot M(y)$ is odd. Therefore, one may apply $z = y^2$ and build a third polynomial in order to extract the additional structure $\{\mu, -\mu\}$.

Equation (14) and (15) enable the computation of the eigenvalues of $P(\lambda)$ to be reduced to solving algebraic equations. From Proposition 3 it follows that possible discrepancies in the Jordan structures can be expected for $y_0 = \pm 2$ and $y_0 = \infty$ corresponding to $\lambda_0 = \pm 1$ and $\lambda_0 = 0, \infty$, respectively.

When $\lambda_0 = \pm 1$ not only the proof we gave is not valid (because, since $w_0 = 0$ is a branch point, there is no neighborhood of analyticity of the matrix function E), but in fact the proposition itself does not hold. As a counterexample, let $a \neq \pm \frac{i}{\sqrt{2}}$ and consider the polynomial

$$P(\lambda) = \begin{pmatrix} \lambda - 2 + \lambda^{-1} & a\lambda - a\lambda^{-1} \\ -a\lambda + a\lambda^{-1} & \lambda + \lambda^{-1} \end{pmatrix}.$$

We have that $\{(1, 0)^T, (0, a)^T\}$ is a Jordan chain for $P(\lambda)$ at $\lambda = 1$. The corresponding $M(y)$ is

$$M(y) = \begin{pmatrix} y - 2 & 0 & 0 & ay^2 - 4a \\ 0 & y & 4a - ay^2 & 0 \\ 0 & a & y - 2 & 0 \\ -a & 0 & 0 & y \end{pmatrix},$$

which has a semisimple eigenvalue at $y = 2$ with the corresponding eigenvectors $(0, 0, 1, 0)^T$ and $(2, 0, 0, a)^T$.

If the leading coefficient of $P(\lambda)$ is not symmetric, then $M(y)$ has $2n$ extra infinite eigenvalues, where n is the dimension of the matrix coefficients of $P(\lambda)$.

These eigenvalues are defective since their geometric multiplicity is only $n + \dim \ker C_{k-1}$, where C_{k-1} is the leading coefficient of $C(y)$.

For the numerical approximation of the roots of $p(y)$ we can exploit again the properties of the Dickson basis to compute the matrix coefficients of $M(y) = \sum_{j=0}^{k+1} M_j \phi_j(y)$. The code below computes the matrices $M_j \in \mathbb{C}^{2n \times 2n}$, $0 \leq j \leq k-1$, given in input the coefficients A_j of $P(\lambda)$, $0 \leq j \leq k$, defined as in (11).

function Dickson_transform

Input: $A_0, \dots, A_k \in \mathbb{C}^{n \times n}$

Output: $M_0, \dots, M_{k+1} \in \mathbb{C}^{2n \times 2n}$

```

     $B_0 = A_0/2; \tilde{C}_0 = 0_n;$ 
    for  $j = 1, \dots, k$ 
         $B_j = (A_j + A_j^T)/2; \hat{C}_j = (A_j - A_j^T)/2;$ 
    end
     $S_0 = 0_n, S_1 = 0_n;$ 
    for  $j = k, \dots, 1$ 
         $S_{\text{mod}(j,2)} = S_{\text{mod}(j,2)} + \hat{C}_j$ 
         $C_{j-1} = S_{\text{mod}(j,2)};$ 
    end
     $C_0 = C_0/2; C_k = C_{k+1} = 0_n; \tilde{C}_0 = C_2$ 
     $\tilde{C}_1 = C_1 + C_3; \tilde{C}_2 = 2C_1 + C_4;$ 
    for  $j = 4, \dots, k$ 
         $\tilde{C}_{j-1} = C_{j-3} + C_{j+1};$ 
    end
     $\tilde{C}_k = C_{k-2}; \tilde{C}_{k+1} = C_{k-1};$ 
    for  $j = 1:k+2$ 
         $\tilde{C}_{j-1} = \tilde{C}_{j-1} - 2C_{j-1};$ 
         $M_{j-1} = [B_{j-1}, \tilde{C}_{j-1}; C_{j-1}, B_{j-1}];$ 
    end
end

```

Remark 4. The coefficients of $C(y)$ are linear combinations of $A_j - A_j^T$. As can be seen by the above algorithm, the coefficients of such combinations expressed in the Dickson basis remain bounded, the upper bound being $1/2$. An analogous result, with upper bound 1, holds for $w^2 C(y)$. This is in contrast with the exponential growth that would have been seen in the purely palindromic case if one had directly applied the Dickson transformation without the use of the Dickson basis.

The arithmetic cost is $\mathcal{O}(n^2 k)$ operations. Once the coefficients M_j are determined, a linearization of $M(y)$ of the form (9) can be constructed. The properties of this linearization are investigated in the next section in order to devise a fast and numerically robust method to evaluate the Newton correction of $p(y)$ defined by (15).

5. Computing the Newton correction

Our aim in this section is to derive a fast, robust and stable method for computing the Newton correction $p(y)/p'(y) = 2 \det(M(y))/(\det(M(y)))'$, where $p(y)$ and $M(y)$ are related by (15), given a structured linearization $L(y) = yE + F$, with $E, F \in \mathbb{C}^{2n(k+1) \times 2n(k+1)}$, of $M(y)$ of the form (9), namely,

$$E = \begin{pmatrix} I_{2n} & & & & \\ & I_{2n} & & & \\ & & \ddots & & \\ & & & I_{2n} & \\ & & & & M_{k+1} \end{pmatrix}$$

and

$$F = \begin{pmatrix} 0 & -2I_{2n} & & & & \\ -I_{2n} & 0 & -I_{2n} & & & \\ & -I_{2n} & 0 & -I_{2n} & & \\ & & \ddots & \ddots & \ddots & \\ & & & -I_{2n} & 0 & -I_{2n} \\ M_0 & M_1 & \dots & M_{k-2} & M_{k-1} & M_k \end{pmatrix}.$$

Our approach relies upon the celebrated *Formula of Jacobi*[13]

$$(\det(L(y)))' = \det(L(y))\text{trace}(L^{-1}(y)L'(y)) = \det(L(y))\text{trace}(L^{-1}(y)E)$$

which reduces the evaluation of $\det(M(y))/(\det(M(y)))' = \det(L(y))/(\det(L(y)))'$ to computing the trace of $L^{-1}(y) \cdot E$. In the sequel we describe a method for finding the block entries and, a fortiori, the trace of the inverse of $L(y)$ from the LQ factorization of the matrix. Then we slightly modify the computation to take into account the contribution due to the matrix E . It will be clear from what follows that this method is general and can be applied, with only trivial modifications, to any kind of unstructured matrix polynomial, simply by considering for instance the standard companion linearization instead of (9).

We denote as $\mathcal{G}(\theta, \psi)$ the 2×2 unitary Givens rotation given by

$$\mathcal{G}(\theta, \psi) = \begin{pmatrix} \theta & \psi \\ -\bar{\psi} & \bar{\theta} \end{pmatrix}, \quad |\theta|^2 + |\psi|^2 = 1.$$

Let $L(y) = \tilde{L} \cdot Q$ be the (block) LQ factorization of $L(y)$ obtained by means of Givens rotations so that

$$L(y)\mathcal{G}_1 \cdot \mathcal{G}_2 \cdots \mathcal{G}_k = \tilde{L}, \quad Q^H = \mathcal{G}_1 \cdot \mathcal{G}_2 \cdots \mathcal{G}_k, \quad (16)$$

$$\mathcal{G}_j = I_{2n(j-1)} \oplus (\mathcal{G}(\theta_j, \psi_j) \otimes I_{2n}) \oplus I_{2n(k-j)}.$$

It can be easily checked that the lower triangular factor \tilde{L} has the following

structure

$$\tilde{L} = \begin{pmatrix} \alpha_1 I_{2n} & & & & & \\ \beta_1 I_{2n} & \alpha_2 I_{2n} & & & & \\ \gamma_1 I_{2n} & \beta_2 I_{2n} & \ddots & & & \\ & \ddots & \ddots & \ddots & & \\ & & \gamma_{k-2} I_{2n} & \beta_{k-1} I_{2n} & \alpha_k I_{2n} & \\ \hat{M}_0 & \hat{M}_1 & \dots & \hat{M}_{k-2} & \hat{M}_{k-1} & \hat{M}_k \end{pmatrix},$$

where $\alpha_j \neq 0$, $1 \leq j \leq k$. If \hat{M}_k and, therefore, $L(y)$ is invertible then the LQ factorization can be used to find a condensed representation of the inverse of $L(y)$. Observe that $L^{-1}(y) = Q^H \cdot \tilde{L}^{-1}$. In order to take into account the occurrence of the matrix E in the Jacobi formula let us introduce the matrix $\tilde{M}_{k+1} = \hat{M}_k^{-1} \cdot M_{k+1}$. Then we have the following

Proposition 4. *There exist matrices $\tilde{M}_1, \dots, \tilde{M}_k \in \mathbb{C}^{2n \times 2n}$ such that*

$$L^{-1}(y)E = \begin{pmatrix} \tilde{M}_1 & \psi_1 \tilde{M}_2 & \dots & \psi_1 \dots \psi_k \tilde{M}_{k+1} \\ & \bar{\theta}_1 \tilde{M}_2 & & \\ & & \ddots & \\ & & & \bar{\theta}_k \tilde{M}_{k+1} \end{pmatrix},$$

where the blank entries are not specified.

Proof. The proof basically follows by applying the (block) Schur decomposition (16) of Q^H to the block lower triangular factor $\tilde{L}^{-1}E$. To show it more formally we can proceed by induction. Let us assume that the the j -th block row of $\mathcal{G}_j \dots \mathcal{G}_k \tilde{L}^{-1}E$ can be represented as

$$\begin{bmatrix} \star & \dots & \star & \tilde{M}_j & \psi_j \tilde{M}_{j+1} & \dots & \psi_j \dots \psi_k \tilde{M}_{k+1} \end{bmatrix},$$

where \tilde{M}_j is the diagonal entry and the value of the entries in the strictly lower triangular part – denoted by \star – is not essential. Then, by applying \mathcal{G}_{j-1} on the left of the matrix we find that the $(j-1)$ -th block row looks like

$$\begin{bmatrix} \star & \dots & \star & \tilde{M}_{j-1} & \psi_{j-1} \tilde{M}_j & \dots & \psi_{j-1} \dots \psi_k \tilde{M}_{k+1} \end{bmatrix},$$

whereas the diagonal entry in position j becomes $\bar{\theta}_{j-1} \tilde{M}_j$. \square

This result says that the block diagonal entries of $L^{-1}(y)$ can be determined from the entries in its first (block) row. The computation of this row is equivalent to the solution of the linear system

$$(I_{2n}, 0_{2n}, \dots, 0_{2n}) = (X_1, \dots, X_{k+1}) \cdot L(y)$$

or, equivalently,

$$(I_{2n}, 0_{2n}, \dots, 0_{2n}) \cdot Q^H = (X_1, \dots, X_{k+1}) \cdot \tilde{L}.$$

In the view of the structure of Q^H this reduces to

$$\left(\theta_1, \theta_2 \psi_1, \dots, \theta_k \prod_{j=1}^{k-1} \psi_j, \prod_{j=1}^k \psi_j \right) \otimes I_{2n} = (X_1, \dots, X_{k+1}) \cdot \tilde{L}.$$

Let $D \in \mathbb{C}^{2n(k+1) \times 2n(k+1)}$ be a block diagonal matrix defined by

$$D = \text{diag}(1, \psi_1, \dots, \prod_{j=1}^{k-1} \psi_j, \prod_{j=1}^k \psi_j) \otimes I_{2n}.$$

Using the matrix D to balance the coefficient matrix yields

$$(\theta_1, \theta_2, \dots, \theta_k, 1) \otimes I_{2n} = (X_1, \dots, X_{k+1}) D^{-1} \cdot D \cdot \tilde{L} \cdot D^{-1}.$$

Observe that

$$\left(\hat{X}_1, \dots, \hat{X}_{k+1} \right) = (X_1, \dots, X_{k+1}) D^{-1} = \left(\tilde{M}_1, \dots, \tilde{M}_{k-1}, \hat{M}_k^{-1} \right),$$

and, therefore, the solution of

$$(\theta_1, \theta_2, \dots, \theta_k, 1) \otimes I_{2n} = \left(\hat{X}_1, \dots, \hat{X}_{k+1} \right) \hat{L}, \quad D \tilde{L} D^{-1} = \hat{L},$$

gives the desired unknown matrices $\tilde{M}_1, \dots, \tilde{M}_k$. To achieve some computational savings we rewrite the system as

$$(\theta_1, \theta_2, \dots, \theta_k, 1) \otimes \hat{M}_k = \left(\tilde{X}_1, \dots, \tilde{X}_{k+1} \right) \hat{L}$$

and thus we arrive at the following relation

$$\det(M(y))' / \det(M(y)) = \text{trace}(\hat{M}_k^{-1}(\tilde{X}_1 + \bar{\theta}_1 \tilde{X}_2 + \dots + \bar{\theta}_{k-1} \tilde{X}_k + \bar{\theta}_k M_{k+1})),$$

which is used to compute the reciprocal of the Newton correction. The function **trace** below implements our resulting algorithm at the cost of $\mathcal{O}(n^2 k + n^3)$ operations.

function trace

Input: $M_0, \dots, M_{k+1} \in \mathbb{C}^{2n \times 2n}$, $\lambda \in \mathbb{C}$, $(\det(M(\lambda)) \neq 0)$

Output: the value of $\eta = p'(\lambda)/p(\lambda)$

$$M_{k-1} = M_{k-1} - M_{k+1}; \quad M_k = M_k + \lambda M_{k+1};$$

$$\alpha = \lambda \text{ ones}(1, k+1); \quad \beta = -\text{ones}(1, k+1);$$

$$\gamma = \text{zeros}(1, k); \quad \chi = -\text{ones}(1, k+1); \quad \chi_1 = -2;$$

for $j = 1, \dots, k$

$$\mathbf{v} = [\alpha_j; \chi_j]; \quad \mathcal{G}^T = \text{planerot}(\mathbf{v}); \quad \mathbf{q}(j, :) = \mathcal{G}(1, :); \quad c_j = \mathbf{q}(j, 1);$$

$$\alpha_j = \alpha_j \mathcal{G}_{1,1} + \chi_j \mathcal{G}_{2,1}; \quad \tilde{\beta} = \beta_j \mathcal{G}_{1,1} + \alpha_{j+1} \mathcal{G}_{2,1};$$

$$\alpha_{j+1} = \beta_j \mathcal{G}_{1,2} + \alpha_{j+1} \mathcal{G}_{2,2}; \quad \beta_j = \tilde{\beta}; \quad \gamma_j = \beta_{j+1} \mathcal{G}_{2,1}; \quad \beta_{j+1} = \beta_{j+1} \mathcal{G}_{2,2};$$

$$M = \mathcal{G}_{1,1} M_{j-1} + \mathcal{G}_{2,1} M_j; \quad M_j = \mathcal{G}_{1,2} M_{j-1} + \mathcal{G}_{2,2} M_j; \quad M_{j-1} = \tilde{M};$$

```

end
for  $j = 1, \dots, k-1$ 
     $\beta_j = \beta_j \mathbf{q}(j, 2);$ 
end
for  $j = 1, \dots, k-2$ 
     $\gamma_j = \gamma_j \mathbf{q}(j, 2) \mathbf{q}(j+1, 2);$ 
end
 $s = 1;$ 
for  $j = k, \dots, 1$ 
     $s = s \mathbf{q}(j, 2); M_{j-1} = s M_{j-1};$ 
end
 $\tilde{X}_k = (c_k M_k - M_{k-1})/\alpha_k; \tilde{X}_{k-1} = (c_{k-1} M_k - M_{k-2} - \beta_{k-1} \tilde{X}_k)/\alpha_{k-1};$ 
for  $j = k-2, \dots, 1$ 
     $\tilde{X}_j = (c_j M_k - M_{j-1} - \beta_j \tilde{X}_{j+1} - \gamma_j \tilde{X}_{j+2})/\alpha_j;$ 
end
 $\tilde{M} = \tilde{X}_1;$ 
for  $j = 1, \dots, k-1$ 
     $\tilde{M} = \tilde{M} + \tilde{c}_j \tilde{X}_{j+1};$ 
end
 $\tilde{M} = \tilde{M} + \tilde{c}_k M_{k+1}; \tilde{M} = M_k \setminus \tilde{M}; \eta = \text{trace}(\tilde{M});$ 

```

6. The Ehrlich-Aberth algorithm for T-palindromic eigenproblems

A simple tool for the simultaneous approximation of all the eigenvalues of a polynomial is the Ehrlich-Aberth method. Bini and Fiorentino [4] showed that a careful implementation of the method yields an efficient and robust polynomial root finder. The software package MPSolve documented in [4] is designed to successfully compute approximations of polynomial zeros at any specified accuracy using a multi-precision arithmetic environment.

A root finder for T-palindromic eigenproblems can be based on the Ehrlich-Aberth method applied for the solution of the algebraic equation $p(y) = 0$, where $p(y)$ is related with $M(y)$ by (15) and $M(y)$ is generated by the function **Dickson_transform** applied to the input coefficients $A_j \in \mathbb{C}^{n \times n}$ of the T-palindromic matrix polynomial $P(\lambda)$ of degree $2k$ given as in (13). The method simultaneously approximates all the zeros of the polynomial $p(y)$: given a vector $\mathbf{z}^{(0)} \in \mathbb{C}^N$, $N = nk$, of initial approximations to the zeros of $p(y)$, the Ehrlich-Aberth iteration generates a sequence $\{\mathbf{z}^{(k)}\}$, $k \geq 0$, which locally converges to the N -tuple of the roots of $p(y)$, according to the equation

$$z_j^{(k+1)} = z_j^{(k)} - \frac{p(z_j^{(k)})/p'(z_j^{(k)})}{1 - \frac{p(z_j^{(k)})}{p'(z_j^{(k)})} \sum_{\ell=1, \ell \neq j}^N \frac{1}{z_j^{(k)} - z_\ell^{(k)}}}, \quad 1 \leq j \leq N.$$

The convergence is superlinear for simple roots and linear for multiple roots. In practice, the Ehrlich-Aberth method exhibits quite good global convergence

properties, even though no theoretical results are known in this regard. The main requirements for an efficient implementation of the method are:

1. a rule for choosing the initial approximations;
2. a fast, numerically robust and stable method to compute the Newton correction $p(z)/p'(z)$;
3. a reliable stopping criterion.

Concerning the first issue it is commonly advocated[3] that for scalar polynomials the convergence benefits from the choice of equally spaced points lying on some circles around the origin in the complex plane. In the case of matrix polynomials where the eigenvalues are often widely varying in magnitude this choice can not be optimal. A better strategy using the initial guesses lying on certain ellipses around the origin in the complex plane is employed in our method. The second task can be accomplished by means of the function **trace** in the previous section. With respect to the third issue, it is worth observing that the QL-based method pursued for the trace computation also provides an estimate on the backward error for the generalized eigenvalue problem. From a result in [14] it follows that if \tilde{y} is not an eigenvalue of $L(y)$ then

$$\eta(\tilde{y}) = 1/(\|(\tilde{y}E + F)^{-1}\|_2 (1 + |\tilde{y}|))$$

gives an appropriate measure of the backward error for the approximate eigenvalue \tilde{y} . Since for $\tilde{y}E + F = \tilde{L} \cdot Q$ we have that

$$\|(\tilde{y}E + F)^{-1}\|_2 = \|L^{-1}\|_2 \geq \|\hat{M}_k^{-1}\|_2 \geq (\sqrt{2n})^{-1} \|\hat{M}_k^{-1}\|_\infty.$$

In our implementation we consider the quantity

$$\hat{\eta}(\tilde{y}) = \sqrt{2n}/(\|\hat{M}_k^{-1}\|_\infty (1 + |\tilde{y}|))$$

as an error measure. If $\hat{\eta}(\tilde{y})$ is smaller than a fixed tolerance then \tilde{y} is taken as an approximate eigenvalue and the corresponding iteration is stopped. The resulting Ehrlich-Aberth algorithm for approximating finite eigenvalues of $M(y)$ and hence obtaining the corresponding eigenvalues of $P(\lambda)$ is described below. In the next section we present results of numerical experiment assessing the robustness of the proposed approach.

function palindromic_aberth_zeros

Input: $A_0, \dots, A_k \in \mathbb{C}^{n \times n}$, $tol \in \mathbb{R}$, $maxit \in \mathbb{N}$, initial guesses z_1, \dots, z_N

Output: approximations $\zeta_1, \dots, \zeta_{2N}$, $N = nk$, of the zeros of $P(\lambda) = \sum_{i=-k}^k A_i \lambda^i$

$[M_0, \dots, M_{k+1}] = \text{Dickson_transform}(A_0, \dots, A_k)$

$N = nk$; $\mathbf{c} = \text{ones}(N, 1)$;

$nn = 0$;

for $i = 1, \dots, maxit$

for $j = 1, \dots, N$

if $(c(j))$

$z = \text{trace}(M_0, \dots, M_{k+1}, z_j)$; $z = 2/z$;

```

        h = sum(1./(z(1:j-1) - z(j)));
        h = h + sum(1./(z(j+1:N) - z(j)));
        h = z/(1 + h*z); z_j = z_j - h;
        if (hat{eta}(z_j) ≤ tol or |h| ≤ tol|z_j|)
            c(j) = 0; nn = nn + 1;
        end;
    end
end
if (nn = N)
    break
end
end
for j = 1, ..., N
    r = roots([1, -z_j, 1]);
    ζ2j-1 = r1; ζ2j = r2;
end

```

The total cost of the algorithm is therefore $\mathcal{O}(t(n^2k + n^3))$ operations, where t is the total number of times that the function **trace** is called. Numerical experiments presented in the next section show that t heavily depends on the choice of the starting points. With a smart choice, t is of order $\mathcal{O}(nk)$, which gives a total computational cost of $\mathcal{O}(n^4k + n^3k^2)$. Since the cost of our method grows as n^4 but is only quadratic in k , where customary QZ-like methods use $\mathcal{O}(n^3k^3)$ operations, an Ehrlich-Aberth approach looks particularly suitable when the matrix polynomial has a high degree and small coefficients so that k^2/n is large.

It is worth noticing that the case of large n can still be treated by means of an Ehrlich-Aberth method in $\mathcal{O}(n^3k^3)$ operations. The basic observation is that the factor n^4 comes from the block structure of the linearization involved in the computation of the trace. A reduction of the cost can therefore be achieved by a different strategy where the linearization is initially converted into (scalar) triangular-Hessenberg form: say, $N(y) = Ry + H$ where R is (scalar) triangular and H is (scalar) Hessenberg. The task can virtually be performed by any extension of the fast structured methods for the Hessenberg reduction proposed in [6, 10]. These methods preserve the rank structure which can therefore be exploited also in the triangular-Hessenberg linearization. Once the matrices R and H have been determined then the computation of $\text{tr}(N(y))$ can be performed by the following algorithm which has a cost of $\mathcal{O}(n^2k^2) = \mathcal{O}(N^2)$ operations:

- Perform a RQ decomposition of the Hessenberg matrix $N(y)$, obtaining a unitary matrix Q represented as product of $\mathcal{O}(N)$ Givens transformations (Schur decomposition) and a triangular matrix U .
- Compute the last row of $N(y)^{-1}R$ by solving $\mathbf{w}^T N(y) = \mathbf{e}_N^T$ and then computing $\mathbf{w}^T := \mathbf{w}^T R$.
- Recover the diagonal entries of $N(y)^{-1}R$ from the entries of \mathbf{w} and the elements of the Schur decomposition of Q .

This alternative road leads to an algorithm of total cost $\mathcal{O}(n^3k^3)$ operations. An efficient implementation exploiting the rank structures of the matrices involved will be presented elsewhere.

7. Numerical Experiments

The function **palindromic_aberth_zeros** for computing the roots of a T-palindromic matrix polynomial $P(\lambda) = \sum_{j=-k}^k A_j \lambda^j$, given its coefficients $A_{-j} = A_j^T \in \mathbb{C}^{n \times n}$, $0 \leq j \leq k$, has been implemented in Matlab¹ and then used for the computation of the zeros of polynomials of both small and high degree. The tolerance is fixed at $tol = 1.e - 13$ and for the maximum number of iterations we set $maxit = 2nk$.

Extensive numerical experiments have been performed to illustrate some basic issues concerned with the efficiency and the accuracy of a practical implementation of our method.

7.1. Efficiency of root-finding

An accurate and efficient root-finder is essential to the success of our algorithm. In practice, the cost of each iteration is strongly dependent on the amount of early convergence (for the sake of brevity, in the following we will refer to this phenomenon using the word *deflation*) occurring for a given problem. In other words, a critical point to assess the efficiency of the novel method is the evaluation of the total number t of calls of the function trace, and of its dependence on the total number $N := nk$ of the eigenvalues. When the Ehrlich-Aberth method is used to approximate scalar polynomials roots, experiments show that t depends on the choice of the starting points. If there is not any a priori knowledge about the location of the roots, empirical evidence [4] shows that choosing starting points distributed on some circles around the origin leads to acceptable performances and/or quite regular convergence patterns.

The class $H_{n,k}$ of T-palindromic polynomials have been used to verify if these properties still hold in the matrix case. The polynomials are constructed according to the following rules:

$$\begin{aligned} H_{n,k} &= \sum_{j=-k}^k A_j \lambda^j, \quad A_j \in \mathbb{R}^{n \times n}, \\ A_0 &= 0_n; \quad A_j = I_n + e_n e_1^T, \quad A_{-j} = A_j^T, \quad 1 \leq j \leq k. \end{aligned}$$

From

$$h(\lambda) = \sum_{j=1}^k \lambda^j + \sum_{j=1}^k \lambda^{-j} = \frac{\lambda^k - 1}{\lambda - 1} \frac{\lambda^{k+1} + 1}{\lambda^k},$$

we find that most of the eigenvalues lie on the unit circle and for k even $\lambda = -1$ is a double root of $h(\lambda)$.

¹Matlab is a registered trademark of The Mathworks, Inc..

Figure 1 describes the convergence history for our root finder applied to $H_{5,20}$ with starting values equally spaced on the circle centered in the origin with radius 4. The curves represented are generated by plotting the sequences $\{z_j^{(k)}\}$, $1 \leq k \leq \text{maxit}$, for $j = 1, \dots, N$. The convergence is quite regular and very similar to that exhibited in the scalar polynomial case [4] and theoretically predicted for simultaneous iterations based on Newton-like methods [17].

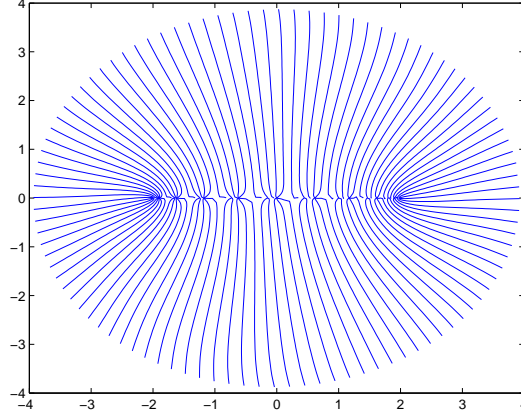


Fig. 1. History of the convergence for the H problem with $n = 5$ and $k = 20$

With this choice of starting points we have observed that the number of global iterations is typically of order of N but there are not enough early deflations, that is, iterations that are prematurely stopped due to early convergence. In order to increase the cost savings due to premature deflation in our program we have employed a slightly refined strategy. Since the method does not approximate directly the eigenvalues λ_i but their Dickson transform $y_i = \lambda_i + \lambda_i^{-1}$, we have chosen starting points on the Dickson transform of the circles $|z| = \rho$, that is points lying on ellipses $\frac{\text{Re}(z)^2}{(\rho+1/\rho)^2} + \frac{\text{Im}(z)^2}{(\rho-1/\rho)^2} = 1$. More precisely, this is the algorithm we used to pick the starting points:

Input: Number N of eigenvalues to approximate and parameters $a \in \mathbb{N}$ and $b \in \mathbb{N}$

Output: Starting points z_k , $k = 1, \dots, N$

```

 $\theta = 2\pi/N;$ 
 $\phi = \text{randn};$ 
for  $j = 1, \dots, N$ 
     $jj = \text{mod}(j, a);$ 
     $\rho = 1 - jj/b;$ 
     $\alpha = \rho + 1/\rho;$ 
     $\beta = 1/\rho - \rho;$ 
     $z_j = \alpha \cos(j * \theta + \phi) + \beta \sin(j * \theta + \phi)$ 

```

end

The integer a determines the number of ellipses whereas b is used to tune the lengths α and β , defined as above, of their semiaxes. We expect that a good choice for the parameters a and b depends on the ratio k/n : when $k \gg n$ we expect many eigenvalues to lie on or near to the unit circle, while when $n \gg k$ we expect a situation more similar to the eigenvalues of a random matrix, with no particular orientation towards unimodularity. We therefore expect that a small ratio a/b works well in the former case while on the contrary in the latter case $a \simeq b$ should be a better choice. Moreover, we expect that as nk grows it is helpful to increase the total number a of ellipses as well.

We show here some of the results on random T-palindromic polynomials. Figure 2 refers to an experiment on small-dimensional, high-degree polynomials: the value of n has been set to 5 while k was variable. The average number of t over a set of 1000 random polynomials for each value of $N = nk$ is shown on the graph. The parameters satisfy $a \in \{2, 3\}$ and $b \in \{8, 64\}$ and they are determined by $a = 1 + 2^c$ and $b = 8^{c+1}$, where the integer c is defined as $c = \log_{320} N$. The graph shows a linear growth of t with respect to $N = nk$.

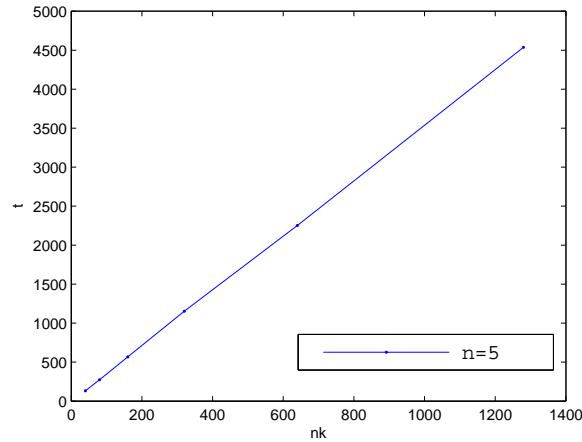


Fig. 2.

Figure 3 refers to an experiment where on the contrary the case of small k is explored. We have considered here $k = 2$ and let n vary and we show the results for t plotted against nk for several choices of a and b . The choice labelled as 'step function' is for $a = \{6, 11\}$ and $b = \{6, 12\}$ generated by $a = 1 + 5 \cdot 2^c$ and $b = 6 \cdot 2^c$. Once again the experiments suggest that when the starting points are conveniently chosen $t \leq \alpha N$ for some constant α and any N in the specified range, and, moreover, the bound still holds for different reasonable choices of the parameters a and b . The experimentation with random polynomials gives $\alpha \simeq 8$ as an estimate for the constant.

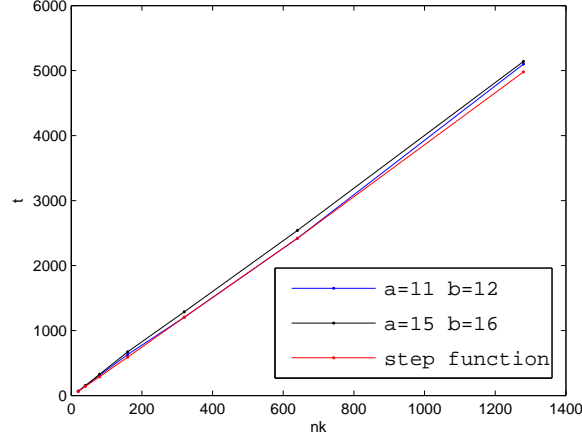


Fig. 3.

In conclusion, the algorithm can greatly benefit from a smart strategy for the selection of the starting points by increasing the number of early deflations. The experiments show that as long as the starting points are suitably chosen the value of t is proportional to $N = nk$.

7.2. Accuracy of root-finding

The other important aspect of our solver based on polynomial root-finding concerns the accuracy of computed approximations. In our experience the method competes very well in accuracy with the customary QZ-algorithm. The accuracy of the computed non-exceptional roots for the random polynomials was always comparable with the accuracy of the approximation obtained with the QZ method. The results of other numerical experiments confirm the robustness of the novel method. Figure 4 illustrates the computed eigenvalues for the problem $H_{5,40}$. Figure 5 also reports the plot of the absolute error vector $abs(\lambda_{EA} - \tilde{\lambda})$ and $abs(\lambda_{QZ} - \tilde{\lambda})$, where $\tilde{\lambda}$ is the vector formed by the eigenvalues computed in high precision arithmetic by Mathematica² while λ_{EA} and λ_{QZ} are, respectively, the vectors formed from the eigenvalues returned by our routine **palindromic_aberth_zeros** and suitably sorted by the internal function *polyeig*.

²Mathematica is a registered trademark of Wolfram Research, Inc.

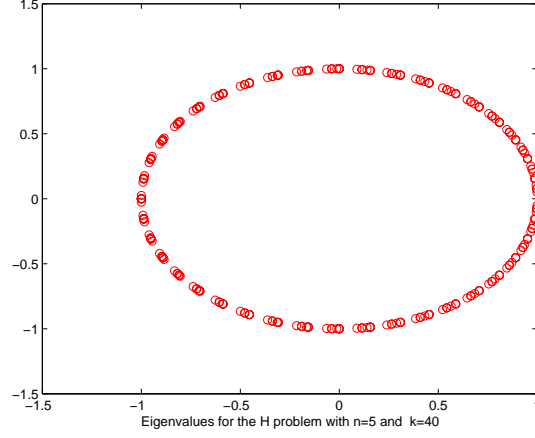


Fig. 4.

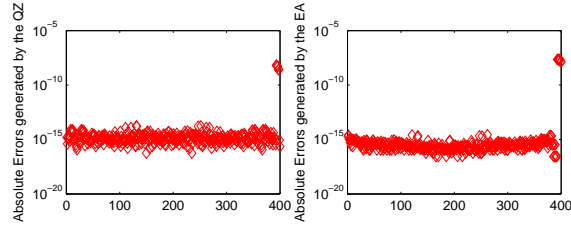


Fig. 5.

The numerical results put in evidence the following important aspects:

1. Poor approximations for the exact eigenvalue $\lambda = -1$ are in accordance with the theoretical predictions: in fact the reverse transformation from $y = \lambda + \lambda^{-1}$ to $\lambda = \frac{1}{2}(y \pm \sqrt{y^2 - 4})$ is known to be ill-conditioned near $y = \pm 2$ (or $\lambda = \pm 1$). Since in this example -1 is a defective eigenvalue, the approximations returned by *polyeig* have comparable absolute errors of order 10^{-8} which are in accordance with the unstructured backward error estimates given in [14].

2. The accuracy of the remaining approximations is unaffected from the occurrence of near-to-critical eigenvalues and is in accordance with the results returned by *polyeig*. For most non-exceptional eigenvalues, the accuracy of approximations computed by our method is slightly better.
3. This kind of behavior is confirmed by many other experiments. Our method performs similarly to the QZ for non-exceptional eigenvalues and for defective exceptional eigenvalues, but generally worse than QZ and the structure-preserving methods [25] for exceptional eigenvalues.

8. Conclusions and future work

In this paper we have shown that the Ehrlich-Aberth method can be used for solving palindromic and T-palindromic generalized eigenproblems. The basic idea can be applied to a generic matrix polynomial of any kind; moreover, as we have shown in this paper, it is possible to adapt it in order to exploit certain structures as the palindromic structure that we have considered here. The resulting algorithm is numerically robust and achieves computational efficiency by exploiting the rank-structure of the associated linearization in the Dickson basis. The algorithm is quite interesting for its potential for parallelization on distributed architectures and, moreover, can be easily incorporated in the MPSolve package to develop a multiprecision root finder for matrix polynomial eigenproblems.

There are, however, some issues that still stand in the way of a fully satisfactory implementation of our method and are currently under investigation.

1. The development of an automatic procedure for the selection of starting points is important to attain a low operation count due to the prevalence and ease of deflation. We have shown that a smart choice could be based on a few parameters to be determined from some rough information on the spectrum localization.
2. The proposed algorithm is still inefficient with respect to the size of the polynomial coefficients. The preliminary reduction of the linearized problem into a Hessenberg-triangular form is the mean to devise a unified efficient algorithm for both small and large coefficients. A fast reduction algorithm would be incorporated in our implementation. The algorithm should be able to exploit the rank structure of the linearization (for large degrees), and, at the same time, the inner structure of the quasiseparable generators (for large coefficients).
3. Regarding the accuracy of the method there are still some difficulties in the numerical treatment of the critical cases. Our current research is focusing on the issue of a structured refinement of the approximations of such eigenvalues.

Acknowledgements. The second author wishes to thank Volker Mehrmann for the many discussions and the valuable comments and suggestions provided during the Gene Golub SIAM Summer School 2010. We are grateful to the anonymous referees for their hints and suggestions.

References

- [1] O. Aberth. Iteration methods for finding all zeros of a polynomial simultaneously. *Math. Comp.*, 27:339–344, 1973.
- [2] A. Amiraslani, R. M. Corless and P. Lancaster. Linearization of matrix polynomials expressed in polynomial bases. *IMA Journal of Numerical Analysis*, 29(1):141–157, 2009.
- [3] D. A. Bini. Numerical computation of polynomial zeros by means of Aberth’s method. *Numer. Algorithms*, 13(2):179–200, 1996.
- [4] D. A. Bini and G. Fiorentino. Design, analysis, and implementation of a multiprecision polynomial rootfinder. *Numer. Algorithms*, 23(2-3):127–173, 2000.
- [5] D. A. Bini, L. Gemignani and F. Tisseur. The Ehrlich-Aberth method for the nonsymmetric tridiagonal eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 27(1):153–175, 2005.
- [6] S. Delvaux and M. Van Barel. A Hessenberg reduction algorithm for rank structured matrices. *SIAM J. Matrix Anal. Appl.*, 29, 895–926, 2007.
- [7] F. De Teran, F. Dopico and D. S. Mackey. Palindromic companion forms for matrix polynomials of odd degree. *Journal of Computational and Applied Mathematics*, 236, 1464–1480, 2011.
- [8] L. E. Dickson. The analytic representation of substitutions on a power of a prime number of letters with a discussion of the linear group. *Ann. of Math.*, 11:65–120, 11:161–183, 1897.
- [9] L. W. Ehrlich. A modified Newton method for polynomials. *Comm. ACM*, 10:107–108, 1967.
- [10] Y. Eidelman, I. Gohberg and L. Gemignani. On the fast reduction of a quasiseparable matrix to Hessenberg and tridiagonal forms. *Linear Algebra Appl.*, 420, 86–101, 2007.
- [11] W. Gander. Zeros of determinants of λ -matrices. In Vadim Olshevsky and Eugene Tyrtyshnikov, editors, *Matrix methods: theory, algorithms and applications, Dedicated to the memory of G. Golub*. World Scientific Publisher, 2010.
- [12] I. Gohberg, P. Lancaster and L. Rodman. *Matrix Polynomials*. Academic Press, New York, 1982.
- [13] M. A. Golberg. The derivative of a determinant. *The American Math. Monthly*, 79:1124–1126, 1972.

- [14] D. J. Higham and N. J. Higham. Structured backward error and condition of generalized eigenvalue problems. *SIAM J. Matrix Anal. Appl.*, 20(2):493–512 (electronic), 1999.
- [15] N. J. Higham, D. S. Mackey, N. Mackey and F. Tisseur. Symmetric linearizations for matrix polynomials. *SIAM Journal of Matrix Analysis and Applications*, 29(1):143–159, 2006.
- [16] M. Hochstenbach, T. Košir and B. Plestenjak. A Jacobi-Davidson type method for a right definite two-parameter eigenvalue problem. *SIAM Journal of Matrix Analysis and Applications*, 26:477–497, 2005.
- [17] J. Hubbard, D. Schleicher and S. Sutherland. How to find all roots of complex polynomials by Newton’s method. *Invent. Math.*, 146:1–33, 2001.
- [18] D. Kressner, C. Schroeder and D. S. Watkins. Implicit QR algorithms for palindromic and even eigenvalue problems. *Numerical Algorithms*, 51(2):209–238, 2009.
- [19] D. S. Mackey, N. Mackey, C. Mehl and V. Mehrmann. Smith forms of palindromic matrix polynomials. *Electron. J. Linear Algebra*, 22, 53–91, 2011.
- [20] D. S. Mackey, N. Mackey, C. Mehl and V. Mehrmann. Structured polynomial eigenvalue problems: good vibrations from good linearizations. *SIAM J. Matrix Anal. Appl.*, 28(4):1029–1051 (electronic), 2006.
- [21] D. S. Mackey, N. Mackey, C. Mehl and V. Mehrmann. Vector spaces of linearizations for matrix polynomials. *SIAM Journal of Matrix Analysis and Applications*, 28(4):971–1004, 2006.
- [22] D. S. Mackey, N. Mackey, C. Mehl and V. Mehrmann. Numerical methods for palindromic eigenvalue problems: Computing the anti-triangular schur form. *Numerical Linear Algebra with Applications*, 16:63–86, 2009.
- [23] C. Mehl. Jacobi-like algorithms for the indefinite generalized Hermitian eigenvalue problem. *SIAM Journal of Matrix Analysis and Applications*, 25:964–985, 2004.
- [24] B. Plestenjak. Numerical methods for the tridiagonal hyperbolic quadratic eigenvalue problem. *SIAM J. Matrix Anal. Appl.*, 28(4):1157–1172 (electronic), 2006.
- [25] C. Schroeder. *Palindromic and Even Eigenvalue Problems - Analysis and Numerical Methods*. PhD thesis, Technischen Universität Berlin, April 2008.
- [26] H. J. S. Smith. On Systems of Linear Indeterminate Equations and Congruences. *Philosophical Transactions of the Royal Society of London*, 151:293–326, 1861.

- [27] G. T. Wilson. The factorization of matricial spectral densities. *SIAM J. Appl. Math.*, 23:420–426, 1972.